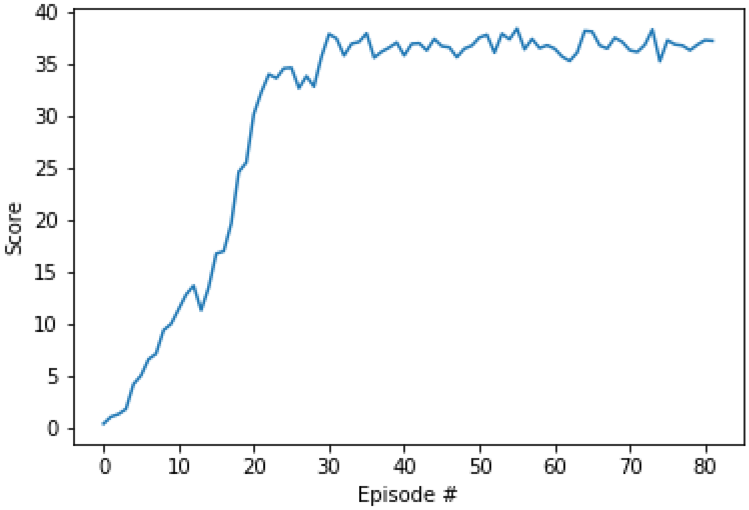
Learning Algorithm:
I was trying to implement Q-Prop, but I faced difficulties translating the algorithm into an actual code due to lack of clarity. Therefore, I implemented the DDPG algorithm.

Hyperparameters:
BUFFER_SIZE = 10000
BATCH_SIZE = 64
GAMMA = 0.99
TAU = 1e-3
LR = 5e-4
UPDATE_EVERY = 4

Model Architecture:
There are two models: actor and critic. The actor takes a state and outputs 64 nodes, which are then taken by another layer of 64 nodes. The output layer consists of action space. Each layer is outputted with relu function except output is through tanh.

The critic on the other hand is more complex and I had to take state size as input. Then, I cat the hidden units of 400 with action size. The output is one decision.

Plot of rewards:
As you see, the algorithm is doing so well it stopped at episode 82 exceeding 30 rewards and averaging around 35 reward points.



Future work:
With reviews from my grader, I plan to continue implementing Q-prop after getting feedback on my current implementation of DDPG. Also, as I packed into future lessons, I could employ MARL for when the environment gets more complicated.